3

orden: 53

Algoritmos y Estructuras de Datos II
Primer parcial – Sábado 4 de mayo de 2019

- El parcial es a **libro abierto**.
- Cada ejercicio debe entregarse **en hojas separadas**.
- Incluir en cada hoja el número de orden asignado, número de hoja, apellido y nombre.
- Antes de entregar, **remover los "pelitos" del borde de las hojas**, si hubiere.
- Cada ejercicio se calificará con **P**erfecto, **A**probado, **R**egular, o **I**nsuficiente.
- El parcial estará aprobado si el ejercicio 1 tiene **A** o **P**, y al menos uno de los dos ejercicios restantes tiene **A** o **P**. Los ejercicios **no** se recuperan por separado.

Ej. 1. Especificación

La Organización de Juegos de Azar, Loterías y Apuestas (OJALÁ) desea implementar un sistema para administrar los sorteos diarios. A lo largo del día, en diferentes puntos de venta ubicados a lo largo de todo el país, cualquier persona mayor de edad puede comprar un ticket de lotería, apostando por una combinación de seis dígitos entre 000000 y 999999. A las 21:00 de cada día se realiza el sorteo, que determina como ganadora a una única combinación. Los ganadores del día reciben un premio que depende del precio del ticket que hayan adquirido y varía de acuerdo con una tabla. Por ejemplo, la tabla podría ser la siguiente:

Precio del ticket	Premio
\$100	\$5.000
\$500	\$50.000
\$1.000	\$200.000

Notar sin embargo que la verdadera tabla aún no está definida y se definirá más adelante. La organización desea conocer en todo momento si el balance es positivo, es decir, si el total de dinero recaudado por la organización supera al total entregado en premios. Además, de acuerdo con disposiciones vigentes, se necesita conocer en todo momento quiénes son los potenciales ludópatas (apostadores compulsivos). En un día dado, una persona se considera potencialmente ludópata si esa persona adquirió más de 50 tickets de lotería en los últimos 100 días. Las personas potencialmente ludópatas no están habilitadas para comprar nuevos tickets.

Notar también que tanto en un mismo día como en días diferentes: (1) varias personas pueden apostar por la misma combinación, (2) una misma persona puede apostar por varias combinaciones, (3) una misma persona puede apostar varias veces por la misma combinación.

Se pide modelar este problema usando TADs.

Ej. 2. Complejidad

Dadas funciones $f, g : \mathbb{N} \to \mathbb{R}_{>0}$ positivas, decidir si las siguientes afirmaciones son verdaderas o falsas. En caso de que sean verdaderas, demostrarlas. En caso de que sean falsas, exhibir un contraejemplo justificando claramente por qué contradice la afirmación.

(a) Si $f(n) \in \Omega(g(n))$ entonces $f(n)^3 \in \Omega(g(n)^3)$.
(b) Si $f(n) \in O(g(n))$ entonces $3^{f(n)} \in O(3^{g(n)})$.

Sea A un arreglo de enteros. Notamos $n = tam(A)$ al tamaño de A. Dado el siguiente algoritmo, escribimos $T_{\text{mejor}}(n)$ para representar su complejidad temporal en mejor caso en función de n, y escribimos $T_{\text{peor}}(n)$ para representar su complejidad temporal en peor caso en función de n.

```
i ← 0
j ← 0
suma ← 0
while i < tam(A) do
    if i == A[i] then
        j ← 0
    end
    while j * j < 64 * i do
        suma ← suma + j * A[i]
        j ← j + 1
    end
    i ← i + 1
end
```

(c) Determinar una $f : \mathbb{N} \to \mathbb{R}_{>0}$ tal que $T_{\text{mejor}}(n) \in \Theta(f(n))$.
(d) Determinar una $g : \mathbb{N} \to \mathbb{R}_{>0}$ tal que $T_{\text{peor}}(n) \in \Theta(g(n))$.

Ej. 3. Diseño

Una red de telefonía móvil cuenta con un conjunto de **antenas** que proveen servicio a distintas **unidades móviles**. Dada una antena y una unidad, se puede saber si la unidad está dentro del radio de alcance de la antena, y en tal caso se puede conocer también la **distancia** de la unidad a la antena, expresada como la cantidad de milisegundos que demora el envío de un mensaje. Si la unidad no está dentro del radio de alcance de ninguna antena, decimos que dicha unidad está **huérfana**. Una unidad que no está huérfana se asigna a la antena más cercana, que llamamos la antena **dueña** de la unidad. En caso de empate entre dos antenas, la unidad se asigna a la antena cuyo identificador sea mayor. Dada una antena a, las unidades cuya madre es a se llaman sus **clientes**. El problema se modela formalmente con el siguiente TAD (incompleto).

TAD Antena **es** Nat
TAD Unidad **es** Nat

TAD Red

géneros red

observadores básicos

antenas : red $\longrightarrow$ conj(antena)

unidades : red $\longrightarrow$ conj(unidad)

alcanza? : red r × antena a × unidad u $\longrightarrow$ bool $\quad\{a \in \text{antenas}(r) \wedge u \in \text{unidades}(r)\}$

distancia : red r × antena a × unidad u $\longrightarrow$ nat $\quad\{(a \in \text{antenas}(r) \wedge u \in \text{unidades}(r)) \wedge_L \text{alcanza?}(a, u)\}$

generadores

...

axiomas

...

Fin TAD

Una red de telefonía móvil se representa con la siguiente estructura:

Red **se representa con** estr

donde estr es tupla ⟨ *tieneDueña?*: dicc(unidad, bool),
huérfanas: conj(unidad),
dueña: dicc(unidad, antena),
distDueña: dicc(unidad, nat),
nroClientes: dicc(antena, nat),
unidadesADistancia: dicc(antena, dicc(nat, conj(unidad))) ⟩

En esta estructura:

- *tieneDueña?* indica si una unidad tiene o no una antena dueña.
- *huérfanas* indica el conjunto de las unidades que están huérfanas, es decir, no tienen antena dueña.
- *dueña* indica la antena dueña de cada unidad no huérfana.
- *distDueña* indica la distancia de una unidad no huérfana a su antena dueña.
- *nroClientes* indica la cantidad de clientes de una antena.
- *unidadesADistancia* indica, dada una antena a y una distancia d, el conjunto de unidades que están a distancia d de la antena a (dentro de su radio de alcance).

Teniendo en cuenta lo descripto arriba se pide:

a) Escribir en castellano el invariante de representación.

b) Escribir formalmente el invariante de representación.

c) Escribir formalmente la función de abstracción.

Martin Ariel [redacted] Orden: 53

Schuster LU: 208/18 Hoja 1

1) TAD día, persona es Nat
TAD ticket es tupla<nat, nat> → numero a apostar → monto ✓

1	2	3
P-	P	P

A

TAD lotería

Observadores:

tabla: lotería ⟶ dicc(nat, nat)

#ticketsXDía: lotería l × día d × persona p → Nat
{$d <$ díaActual(l) ∧ $d >$ díaActual(l) − 100}

ticketsXPersona: lotería l × persona p → multiconj(ticket)
{$p \in$ compraron(l)}

compraron: lotería → conj(persona)

díaActual: lotería → Nat

balanceActual: lotería → Nat   Debería ser un int.

Generadores:

iniciar: dicc(nat, nat) d ⟶ lotería {¬∅?(claves(d))}

comprarTicket: lotería l × persona p × Nat n × Nat monto → lotería
~~× nat monto~~
{$0 \leq n$ ∧ $n \leq 999999$ ∧
monto ∈ claves(tabla(l)) ∧
¬ ludoPata?(l, p)}

sortear: lotería l × Nat N → lotería
{$0 \leq N$ ∧ $N \leq 999999$}

Otras Operaciones:

premios: lotería l × nat n × conj(persona) P_s → lotería {$P_s \subseteq$ ~~persona~~ compraron(l)}
ludoPata?: lotería l × persona p → bool
balancePos?: lotería → bool
~~...~~
premiosXPersona: dicc(nat, nat) × nat, multiconj(tickets) → Nat
ludoPataAux: lotería l × persona × nat × nat → bool

Axiomas:

tabla(iniciar(d)) ≡ d

tabla(comprarTicket(l, P, N, M)) ≡ tabla(l)

tabla(sortear(l, N)) ≡ tabla(l)

#ticketsXDia(iniciar(d), d, P) ≡ 0

#ticketsXDia(comprarTicket(l, P, N, M), d, P) ≡ #ticketsXDia(l, d, P)

#ticketsXDia(sortear(l, N), d, P) ≡ ~~IF d = diaActual(l) then~~ ~~#ticketsXPersona~~

IF d = diaActual(l) ∧ P ∈ compraron(l)

then #ticketsXPersona(l, P)

else IF d = diaActual(l) then 0

else #ticketsXDia(l, d, P)

Fi

Fi

~~ticketsXPersona(comprar~~

ticketsXPersona(comprarTicket(l, P, N, M), P') ≡

IF P = P' then Ag(⟨N, M⟩, ticketsXPersona(l, P))

else ticketsXPersona(l, P')

Fi

~~ticketsXPersona~~

diaActual(iniciar(d)) ≡ 0

diaActual(comprarTicket(l, P, N, M)) ≡ diaActual(l)

diaActual(sortear(l, N)) ≡ diaActual(l) + 1

Martin Ariel Schuster [redacted]
LU: 208/18

orden: 53
Hoja 2

$compraron(iniciar(d)) \equiv \emptyset$
$compraron(comprarTicket(\ell, P, N, M)) \equiv \{P\} \cup compraron(\ell)$
$compraron(sortear(\ell, N)) \equiv \emptyset$

$balanceActual(iniciar(d)) \equiv 0$
$balanceActual(comprarTicket(\ell, P, N, M)) \equiv balanceActual(\ell) + M$
$balanceActual(sortear(\ell, N)) \equiv balanceActual(\ell) - premios(\ell, N, compraron(\ell))$

$premios(\ell, n, Ps) \equiv$ if $\emptyset?(Ps)$ then 0
else $premiosXPersona(tabla(\ell), n, ticketsXPersona(\ell, dameUno(Ps))) +$
$premios(\ell, n, sinUno(Ps))$
fi

$premiosXPersona(t, N, ts) \equiv$ if $\emptyset?(ts)$ then 0
else if $\pi_1(dameUno(ts)) = N$
then $obtener(\pi_2(dameUno(ts)), t) + premiosXPersona(t, N, sinUno(ts))$
else $premiosXPersona(t, N, sinUno(ts))$
fi
fi

$ludopata?(\ell, P) \equiv ludopataAux(\ell, P, diaActual(\ell) - 1, 0)$
$ludopataAux(\ell, P, d, ts) \equiv$ if $d = diaActual(\ell) - 101 \lor d = 0$ then false
else if $ts + \#ticketsXDia(\ell, d, P) \geq 50$ then true
else $ludopataAux(\ell, P, d-1, ts + \#ticketsXDia(\ell, d, P))$
fi
fi

$balancePos?(\ell) \equiv balanceActual(\ell) > 0$

Igualdad Observacional:

$(\forall l_1, l_2 : loteria)(l_1 =_{obs} l_2 \longleftrightarrow$

$tabla(l_1) =_{obs} tabla(l_2) \land diaActual(l_1) =_{obs} diaActual(l_2) \land$

$balanceActual(l_1) =_{obs} balanceActual(l_2) \land compraron(l_1) =_{obs} compraron(l_2) \land_L$

$(\forall p : persona)(\forall d : dia)(d \geq 0 \land d < diaActual(l_1) \longrightarrow_L$

$\#ticketsXDia(l_1, d, p) =_{obs} \#ticketsXDia(l_2, d, p)) \land$

$(\forall p : persona)(p \in compraron(l_1) \longrightarrow_L$

$ticketsXPersona(l_1, p) =_{obs} ticketsXPersona(l_2, p))))$

Decisiones y Aclaraciones:

- No me interesa cómo está definida la tabla, solo quiero que esté definida. (consultado a Pablo)
- Si bien el enunciado dice que quiero saber si una persona es ludópata "dado un día", puede no hacerse con un día particular sino siempre desde el día actual. Considero a una persona ludópata desde el día anterior al actual (consultado a Pablo)
- Una persona puede comprar muchos tickets pero las compras son individuales.
  i.e.: si quiere comprar N tickets, el vendedor deberá comprar ~~[illegible]~~ N veces c/v. (consultado a Pablo)

OK.

Martín Ariel Schuster [redacted] LU: 208/18

orden 53
Hoja 3

HOJA N°
FECHA

2) Si $f(n) \in \Omega(g(n))$, entonces $f(n) \geq c \cdot g(n)$. Verdadero

$$\longleftrightarrow g(n)^3 \cdot \underbrace{c^3}_{k} \leq f(n)^3 \longleftrightarrow g(n)^3 k \leq f(n)^3$$

elevar al cubo es creciente y $f, g : \mathbb{N} \to \mathbb{N}$

luego, $f(n)^3 \in O(g(n)^3)$

b) Falso, basta ver que tomando $f(n) = kn$ y $g(n) = n$, se cumple que $f(n) \in O(g(n))$, luego, vemos que $3^{kn} \leq 3^n c' \longleftrightarrow 3^{kn-n} \leq c'$

$\longleftrightarrow 3^{n(k-1)} \leq c$, en part para $k=2$

$\longleftrightarrow 3^n \leq c$ Abs!

d) El peor caso de este algoritmo sería aquel en el cual $(\forall i : nat)(0 \leq i < tam(A) \Rightarrow i = A[i])$ Puesto que de esta manera j siempre se reestablecería en 0. (si no se cumpliese, habría iteraciones de i donde j no se resetearía y se iteraría menos veces el otro while)

Luego $T_{peor}(n) = \sum_{i=0}^{N} (OE + \sum_{j=0}^{8\sqrt{i}} OE) = \sum_{i=0}^{N} (OE + 8\sqrt{i}) = N + \sum_{i=0}^{N} 8\sqrt{i}$

operaciones elementales

Luego tomo $f(n) = N + \sum_{i=0}^{N} 8\sqrt{i}$.

Creo que es $N\sqrt{N}$ pero prefiero notarlo de esta manera y que sea esta mi respuesta.

NOTA

c) El mejor caso sería aquel donde
$\forall i.nat \quad 0 \leq i < tam(A) \longrightarrow A[i] \neq i$, luego
j nunca se reiniciaría y el while de abajo se ejecutaría
muchas menos veces.
Si en algún i, $A[i]=i$, j se reiniciaría y se
iteraría más veces.

Luego, como j nunca se reinicia, el while de
abajo iterará a lo sumo $\sqrt{N}$ veces, luego

$T_{mejor}(n) = N+\sqrt{N}$ y basta tomar
$g(N) = N+\sqrt{N}$

considerando que las
operaciones elementales
son $O(1)$

(ADEMÁS $g \in \Theta(n)$).

Orden 53

Hoja 4

Martín Ariel [redacted]

Schuster LU: 208/18

3) a)

Rep:

1) Todas las unidades a las que se haga referencia deben estar en claves(e. tieneDueño?)

2) ~~[illegible]~~

Una unidad tiene dueño sii está en claves (e. dueño)

3) Una unidad no tiene dueño sii está en e. huerfanas

4) Las unidades que tienen dueño deben distar de ella y viceversa

5) Las antenas dueñas de alguna unidad ∈ claves (e. nro clientes) = claves(e.unidADist)

6) Una antena es dueña de una unidad ⟷ está a alguna distancia de la antena y es la mínima respecto al resto de las antenas. (o es igual y el nº de antena es Mayor)

7) La distancia a dueño de una unidad es la distancia de una unidad a su antena dueña.

8) El nro de clientes es la #unidades de las que es dueña esa antena.

9) No hay 2 unidades a 2 distancias distintas de una misma antena.

ESTO PODRÍA PASAR.

$Rep : \widehat{estr} \longrightarrow boolean \quad , \quad \forall e : \widehat{estr}$

$Rep(e) \equiv 1 \land 2 \land 3 \land 4 \land 5 \land_L (6 \land 7 \land_L (8 \land 9))$

1) $claves(e.dueño) \subseteq claves(e.tieneDueño?) \land e.huerfanos \subseteq claves(e.tieneDueño?)$

2) $(\forall u : unidad)(def?(u, e.tieneDueño?) \rightarrow_L obtener(u, e.tieneDueño?) \leftrightarrow u \in claves(e.dueño))$

3) $(\forall u : unidad)(def?(u, e.tieneDueño?) \rightarrow_L \neg obtener(u, e.tieneDueño?) \leftrightarrow u \in e.huerfanos)$

4) $claves(e.dueño) = claves(e.distDueño) \land$

$(\forall a : antena)(a \in claves(e.unidADist) \rightarrow_L$
$(\forall n : nat)(n \in claves(obtener(a, e.unidADist)) \rightarrow_L$
$obtener(n, obtener(a, e.unidADist)) \subseteq claves(e.tieneDueño)))$

5) $claves(enroClientes) = claves(e.unidADist) \land$

$(\forall u : unidad)(u \in claves(e.dueño) \rightarrow_L obtener(u, e.dueño) \subseteq claves(e.nroClientes))$

6) $(\forall u : unidad)(def?(u, e.dueño) \rightarrow_L$
$(\exists a : antena)(def?(a, e.unidADist) \land_L (\exists n : nat)(def?(n, obtener(a, e.unidADist))$
$\land_L u \in obtener(n, obtener(a, e.unidADist)) \land$
$\neg\left[(\exists a' : antena)(\neg a = a' \land def?(a, e.unidADist) \land_L (\exists n' : nat)(def?(n, obtener(a, e.unidADist)) \land_L \right.$
$\left. u \in obtener(n, obtener(a, e.unidADist)) \land (n' < n \lor n = n' \land a' > a)))\right]$
$\land a = obtener(u, e.dueño)))) \land$

$(\forall a : antena)(def?(a, e.unidADist) \rightarrow_L (\forall n : nat)(def?(n, obtener(a, e.unidADist))$
$\rightarrow_L obtener(n, obtener(a, e.unidADist)) \subseteq claves(e.dueño)))$

Martín Ariel [redacted] Schuster   LU: 208/18   Hoja 5

7) $(\forall u : \text{unidad})(\text{def?}(u, e.\text{distDueño}) \Rightarrow_L$

~~obtener(u, e.distDueño)~~ .

$(\exists n : \text{nat})(\text{def?}(n, \text{obtener}(\text{obtener}(u, e.\text{dueño}), e.\text{unidADist})) \land_L$

~~u ∈ obtener(n, obtener(obtener(u, e.dueño), e.unidADist)~~

$u \in \text{obtener}(n, \text{obtener}(\text{obtener}(u, e.\text{dueño}), e.\text{unidADist})) \land$

$n = \text{obtener}(u, e.\text{distDueño})))$

8) $(\forall a : \text{antena})(\text{def?}(a, e.\text{nroClientes}) \Rightarrow_L$

$\text{obtener}(a, e.\text{nroClientes}) = \#\text{unidAnt}(a, e.\text{dueño}, \text{claves}(e.\text{dueño})))$

9) $(\forall a : \text{antena})(\text{def?}(a, e.\text{unidADist}) \Rightarrow_L$

$(\forall n, n' : \text{nat})(\text{def?}(n, n', \text{obtener}(a, e.\text{unidADist})) \land \neg n = n' \Rightarrow_L$

$\emptyset?(\text{obtener}(n, \text{obtener}(a, e.\text{unidADist})) \cap \text{obtener}(n', \text{obtener}(a, e.\text{unidADist})))))$

$\#\text{unidAnt} : \text{antena} \times \text{dicc}(\text{unidad}, \text{antena}) \times \text{conj}(\text{unidad}) \rightarrow \text{nat}$

$\#\text{unidAnt}(a, d, us) \equiv$ if $\emptyset?\,us$ then 0 else

(if $a = \text{obtener}(d, \text{dameUno}(us))$ then 1 else 0 fi) +

$\#\text{unidADist}(a, d, \text{sinUno}(us))$

fi